Multiple Linear Regression

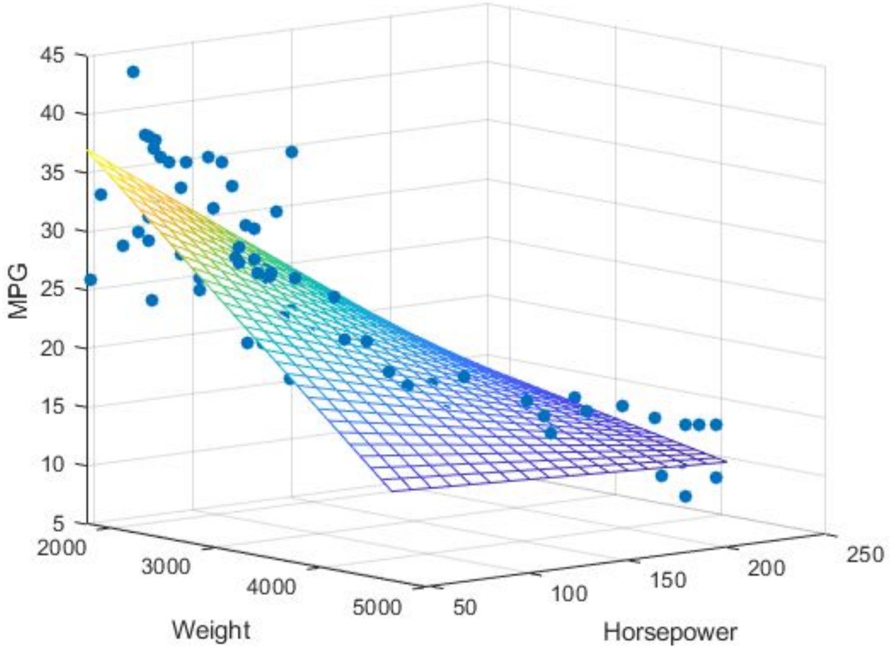
// FLATIRON SCHOOL

Multiple Linear Regression

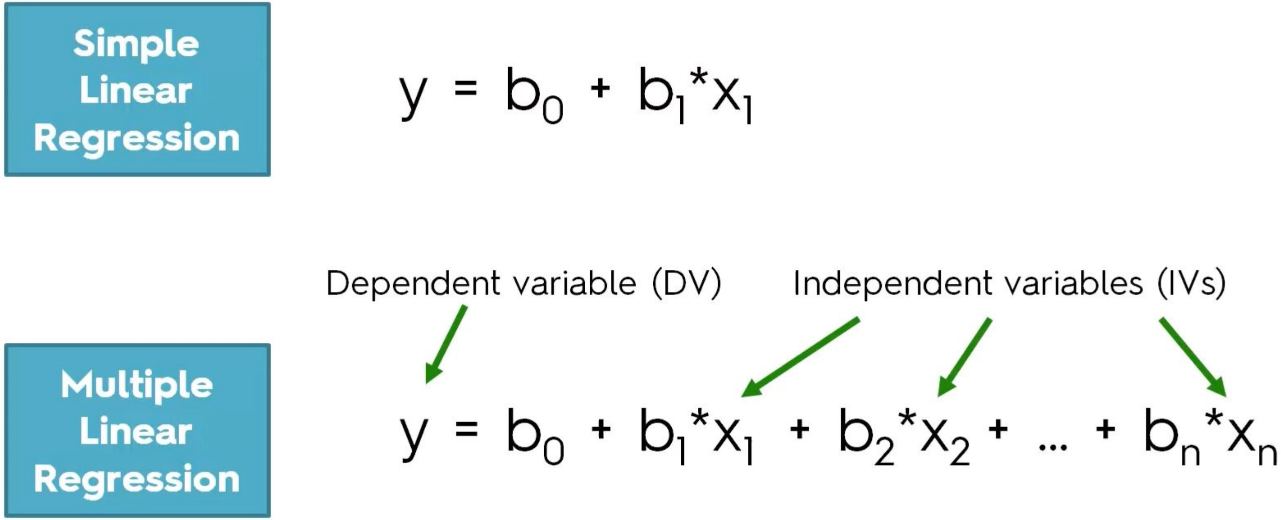
- Equation of the Line
- Regression Coefficients
- Matrix Approach
- The Least Squares
- The Coefficient of Determination
- Assumptions of the Regression

Multiple Linear Regression

- several explanatory variables to predict an outcome
- an extension of linear (OLS) regression modeling
- continuous outcome variable
  - continuous or categorical predictors



Multiple Linear Regression



Simple Linear Regression

$$y = b_0 + b_1 * x_1$$

Multiple Linear Regression

Dependent variable (DV)

Independent variables (IVs)

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + \ldots + b_n * x_n$$

Equation of the Line- Least Squares

$$
\begin{aligned}
\varepsilon'\varepsilon &= (y - X\hat{\beta})'(y - X\hat{\beta}) \\
&= y'y - 2\hat{\beta}'X'y + \hat{\beta}'X'X\hat{\beta} \\
&= y'y - 2\hat{\beta}'X'y + \hat{\beta}'X'X[X'X]^{-1}X'y \\
&= y'y - 2\hat{\beta}'X'y + \hat{\beta}X'y \\
&= y'y - \hat{\beta}'X'y
\end{aligned}
$$

$$\varepsilon'\varepsilon = y'y - \hat{\beta}'X'y$$

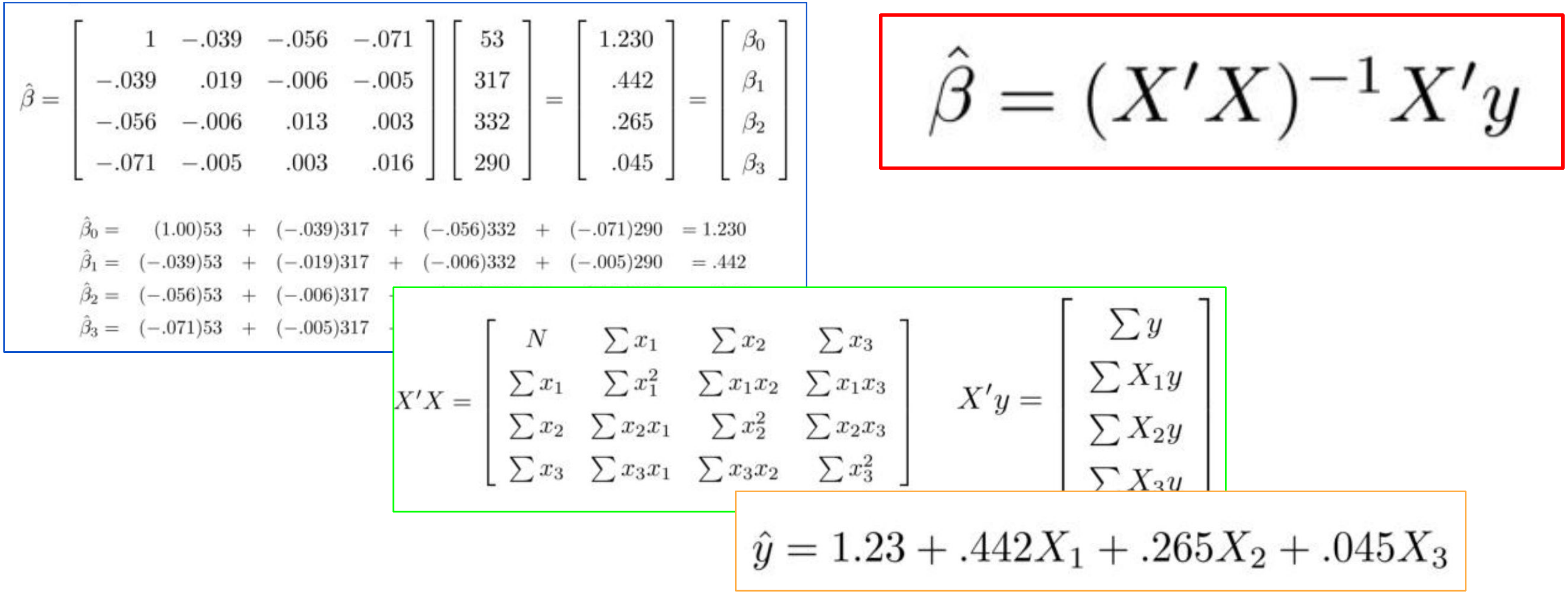
$$y'y - n\bar{y}^2 = (\hat{\beta}'X'y - n\bar{y}^2) + \varepsilon'\varepsilon$$

$$\underbrace{y'y - n\bar{y}^2}_{\boxed{SS_{tot}}} = \underbrace{(\hat{\beta}'X'y - n\bar{y}^2)}_{\boxed{SS_{reg}}}+$$

$$
\begin{cases}
SS_{reg} &= \hat{\beta}'X'y - \dfrac{(\sum y)^2}{n} \\
SS_{res} &= y'y - \hat{\beta}'X'y \\
SS_{tot} &= y'y - n\bar{y}^2
\end{cases}
$$

$$SS_{tot} = SS_{res} + SS_{reg}$$

Equation of the Line- Matrix Approach



$$\hat{\beta} = \begin{bmatrix} 1 & -.039 & -.056 & -.071 \\ -.039 & .019 & -.006 & -.005 \\ -.056 & -.006 & .013 & .003 \\ -.071 & -.005 & .003 & .016 \end{bmatrix} \begin{bmatrix} 53 \\ 317 \\ 332 \\ 290 \end{bmatrix} = \begin{bmatrix} 1.230 \\ .442 \\ .265 \\ .045 \end{bmatrix} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}$$

$$\begin{aligned} \hat{\beta}_0 &= (1.00)53 + (-.039)317 + (-.056)332 + (-.071)290 = 1.230 \\ \hat{\beta}_1 &= (-.039)53 + (-.019)317 + (-.006)332 + (-.005)290 = .442 \\ \hat{\beta}_2 &= (-.056)53 + (-.006)317 \\ \hat{\beta}_3 &= (-.071)53 + (-.005)317 \end{aligned}$$

$$\hat{\beta} = (X'X)^{-1}X'y$$

$$X'X = \begin{bmatrix} N & \sum x_1 & \sum x_2 & \sum x_3 \\ \sum x_1 & \sum x_1^2 & \sum x_1x_2 & \sum x_1x_3 \\ \sum x_2 & \sum x_2x_1 & \sum x_2^2 & \sum x_2x_3 \\ \sum x_3 & \sum x_3x_1 & \sum x_3x_2 & \sum x_3^2 \end{bmatrix} \qquad X'y = \begin{bmatrix} \sum y \\ \sum X_1y \\ \sum X_2y \\ \sum X_3y \end{bmatrix}$$

$$\hat{y} = 1.23 + .442X_1 + .265X_2 + .045X_3$$

Goodness of Fit - Coefficient of Determination

$$Coefficient\ of\ Determination \rightarrow \quad R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

$$Sum\ of\ Squares\ Total \rightarrow \quad SST = \sum (y - \bar{y})^2$$

$$Sum\ of\ Squares\ Regression \rightarrow \quad SSR = \sum (y' - \bar{y'})^2$$

$$Sum\ of\ Squares\ Error \rightarrow \quad SSE = \sum (y - y')^2$$

- **R^2 = variance of the dependent variable explained by the regression model**
- **Perfect fit, SSE = 0, and R^2 is 1; total failure, SSE = SST, R^2 = 0**

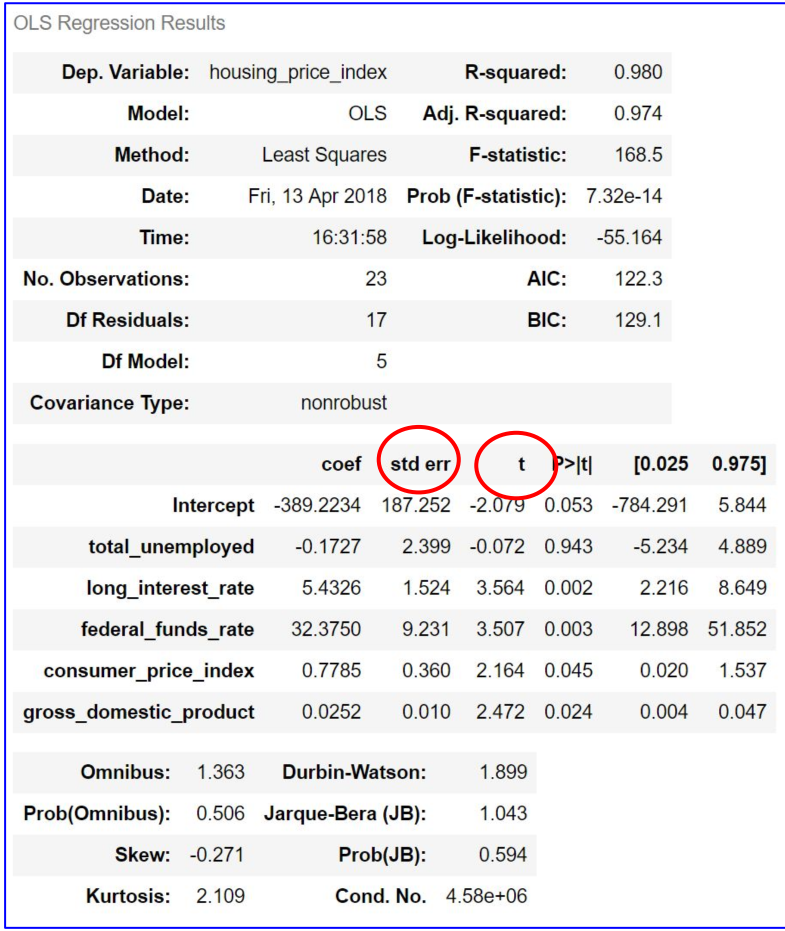
Significance of the Model

Significance test →

$$t_{(n-p-1)} = \frac{b_i}{S_e(b_i)} = \frac{b_i}{\sqrt{S_e^2(\mathbf{X}'\mathbf{X})^{-1}}}$$

Standard Error of Estimation →

$$S_e = \sqrt{\frac{SSE}{n-p-1}}$$



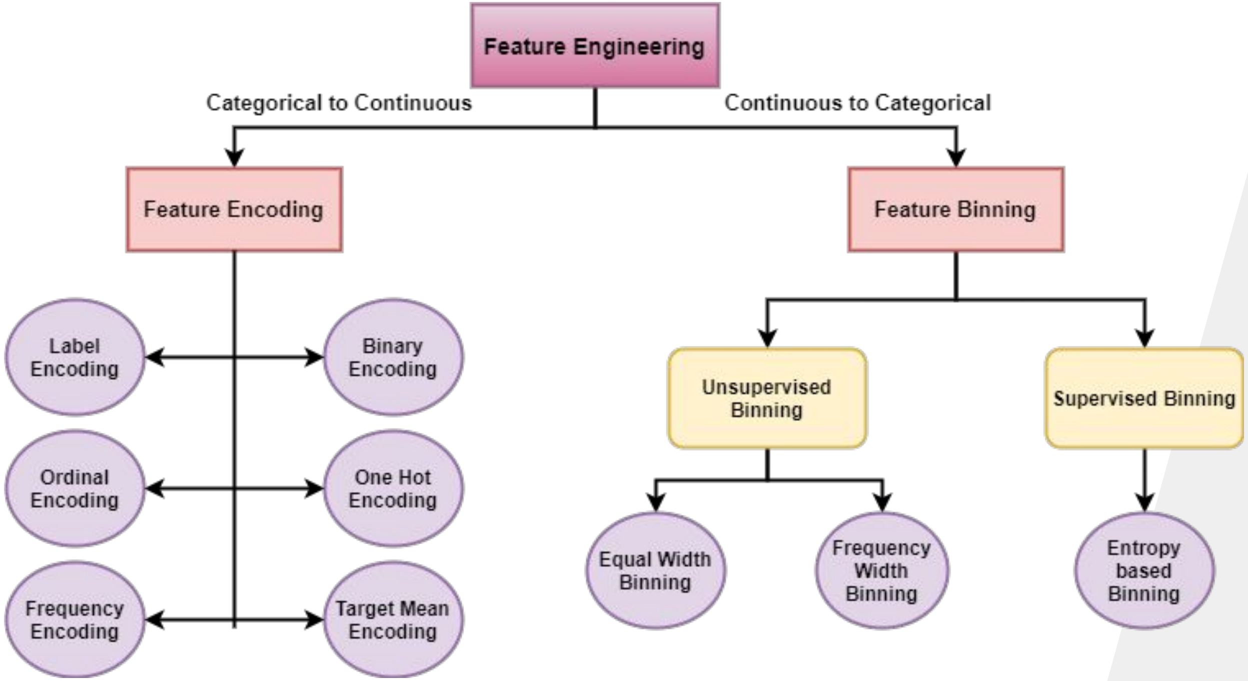
OLS Regression Results

Dep. Variable:	housing_price_index	**R-squared:**	0.980
Model:	OLS	**Adj. R-squared:**	0.974
Method:	Least Squares	**F-statistic:**	168.5
Date:	Fri, 13 Apr 2018	**Prob (F-statistic):**	7.32e-14
Time:	16:31:58	**Log-Likelihood:**	-55.164
No. Observations:	23	**AIC:**	122.3
Df Residuals:	17	**BIC:**	129.1
Df Model:	5		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-389.2234	187.252	-2.079	0.053	-784.291	5.844
total_unemployed	-0.1727	2.399	-0.072	0.943	-5.234	4.889
long_interest_rate	5.4326	1.524	3.564	0.002	2.216	8.649
federal_funds_rate	32.3750	9.231	3.507	0.003	12.898	51.852
consumer_price_index	0.7785	0.360	2.164	0.045	0.020	1.537
gross_domestic_product	0.0252	0.010	2.472	0.024	0.004	0.047

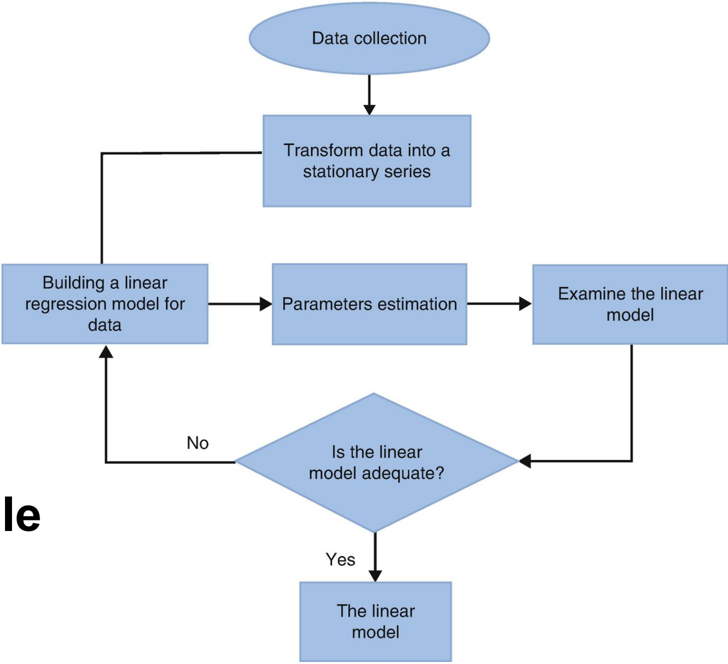
Omnibus:	1.363	**Durbin-Watson:**	1.899
Prob(Omnibus):	0.506	**Jarque-Bera (JB):**	1.043
Skew:	-0.271	**Prob(JB):**	0.594
Kurtosis:	2.109	**Cond. No.**	4.58e+06

Categorical vs. Continuous Variables

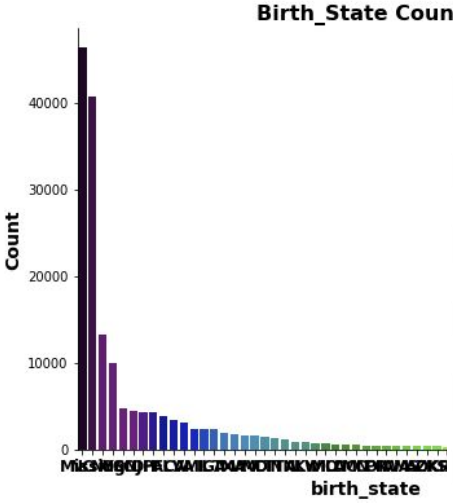


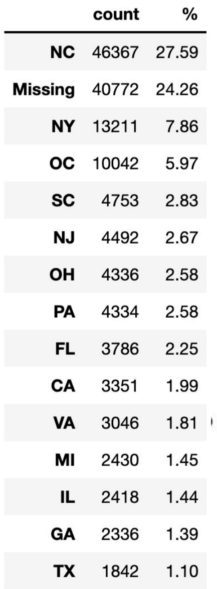
Basic Model Selection

- **drop insignificant coefficients**
  - reduce the average error of predictions
- **estimation of regression coefficients unstable**
  - due to multicollinearity
- **a simpler model is a better model**
  - more insight into the influence of predictors

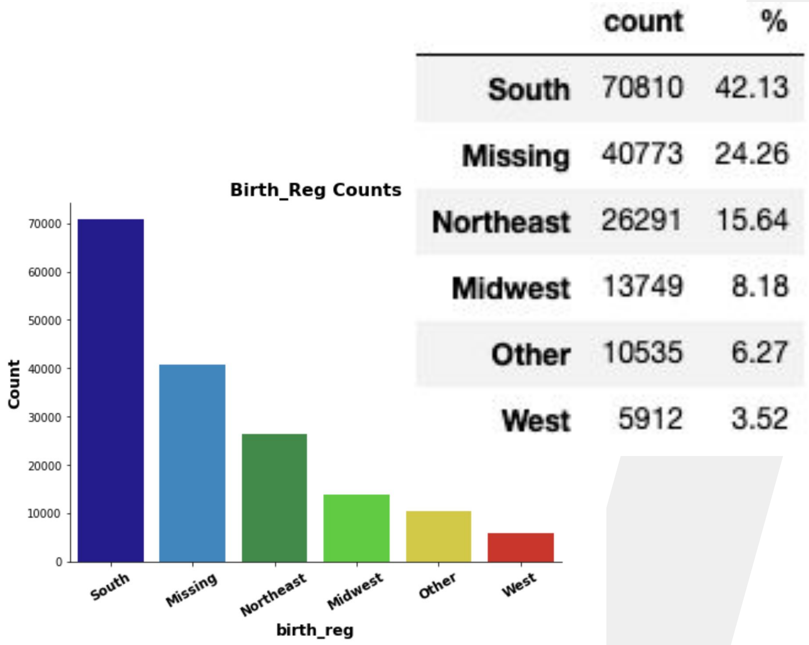

Data collection
Transform data into a stationary series
Building a linear regression model for data
Parameters estimation
Examine the linear model
Is the linear model adequate?
No
Yes
The linear model

Binning Categorical Variables



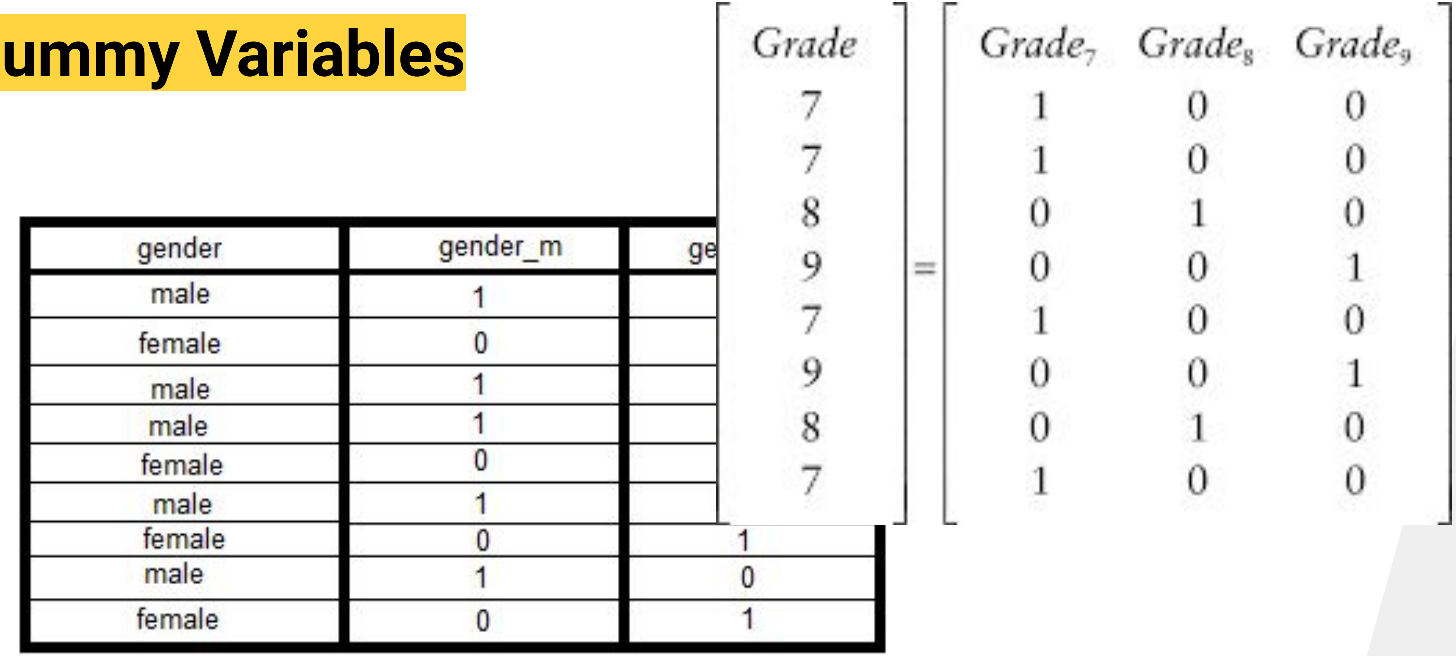


	count	%
NC	46367	27.59
Missing	40772	24.26
NY	13211	7.86
OC	10042	5.97
SC	4753	2.83
NJ	4492	2.67
OH	4336	2.58
PA	4334	2.58
FL	3786	2.25
CA	3351	1.99
VA	3046	1.81
MI	2430	1.45
IL	2418	1.44
GA	2336	1.39
TX	1842	1.10



	count	%
South	70810	42.13
Missing	40773	24.26
Northeast	26291	15.64
Midwest	13749	8.18
Other	10535	6.27
West	5912	3.52

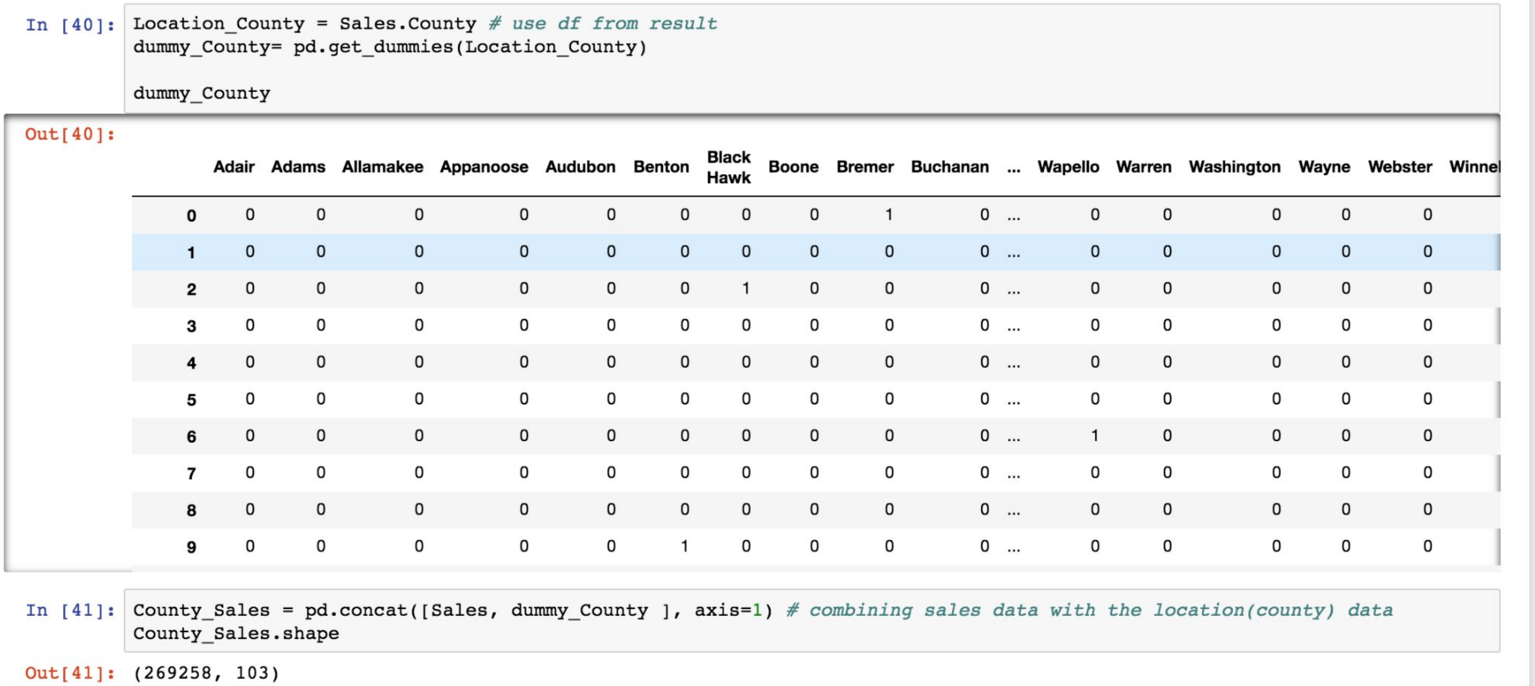
Dummy Variables



gender	gender_m	ge
male	1	
female	0	
male	1	
male	1	
female	0	
male	1	
female	0	1
male	1	0
female	0	1

$$\begin{bmatrix} Grade \\ 7 \\ 7 \\ 8 \\ 9 \\ 7 \\ 9 \\ 8 \\ 7 \end{bmatrix} = \begin{bmatrix} Grade_7 & Grade_8 & Grade_9 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Dummy Variables



```
In [40]: Location_County = Sales.County # use df from result
         dummy_County= pd.get_dummies(Location_County)

         dummy_County
```

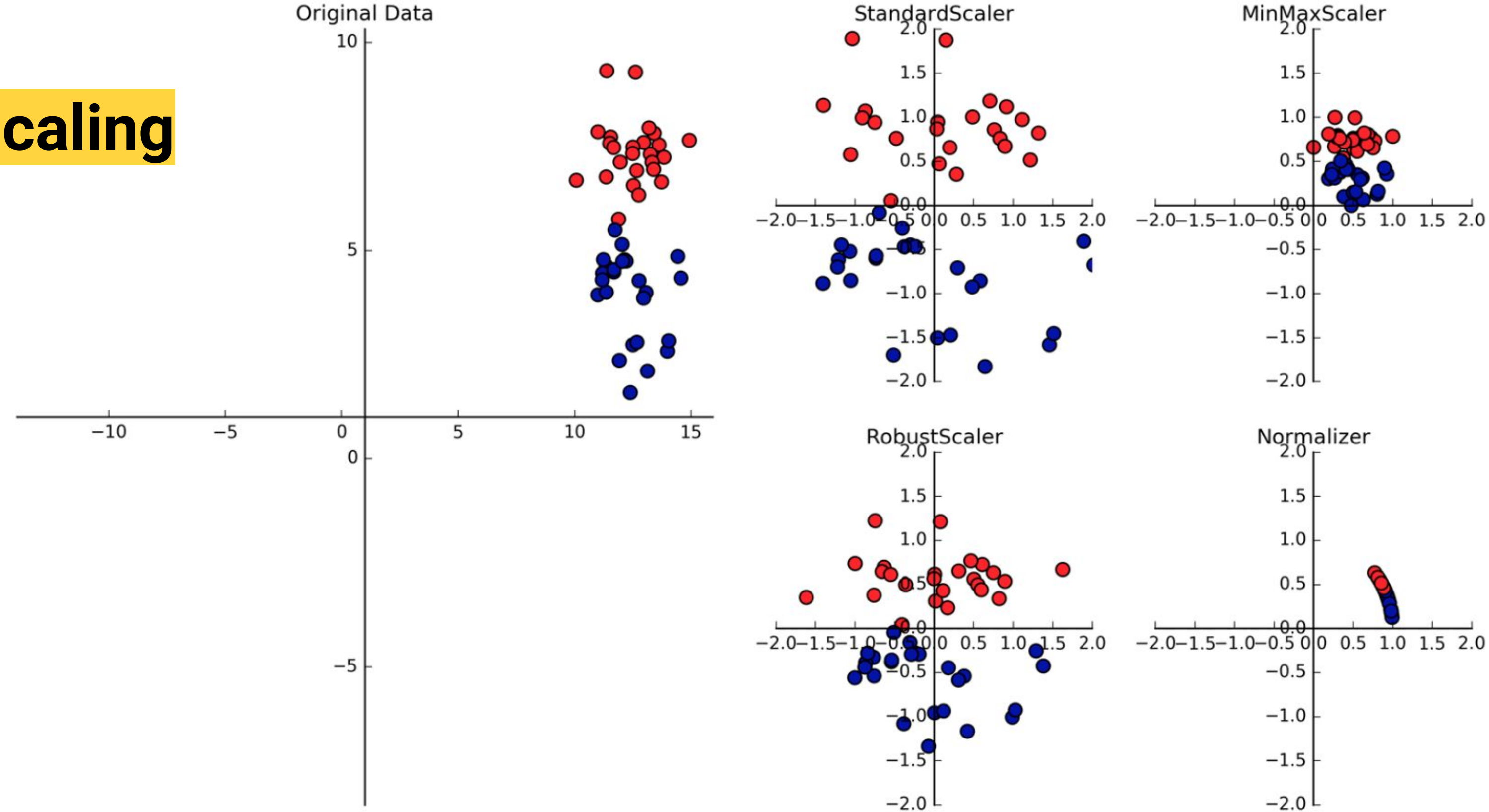
Out[40]:

	Adair	Adams	Allamakee	Appanoose	Audubon	Benton	Black Hawk	Boone	Bremer	Buchanan	...	Wapello	Warren	Washington	Wayne	Webster	Winnel
0	0	0	0	0	0	0	0	0	1	0	...	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
2	0	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	0	...	1	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
9	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	

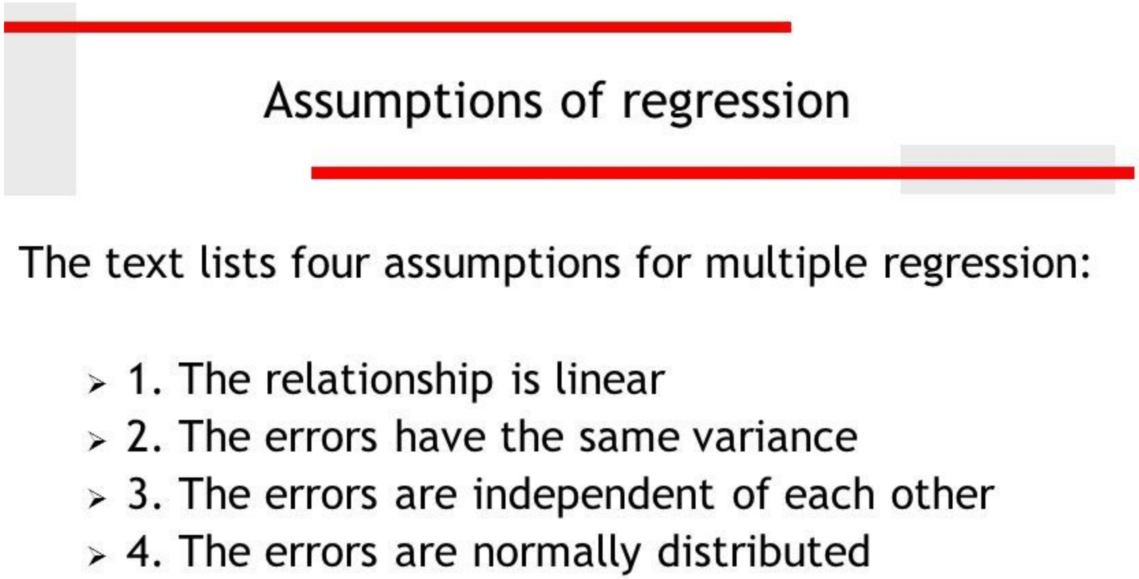
```
In [41]: County_Sales = pd.concat([Sales, dummy_County ], axis=1) # combining sales data with the location(county) data
         County_Sales.shape

Out[41]: (269258, 103)
```

Feature Scaling



Assumptions of Multiple Linear Regression



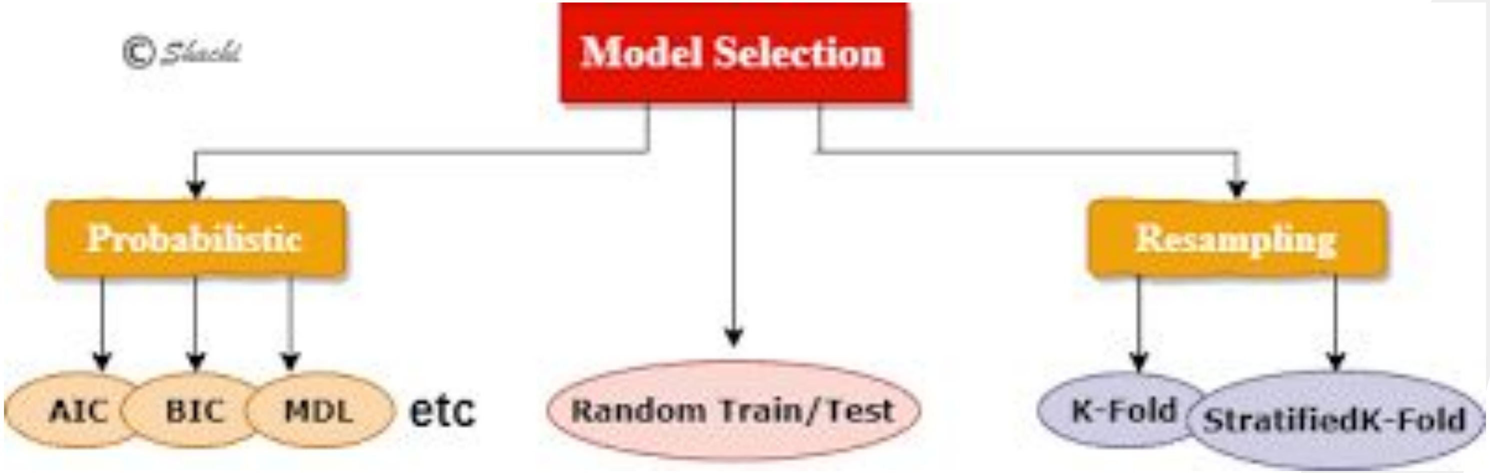
Assumptions of regression

The text lists four assumptions for multiple regression:

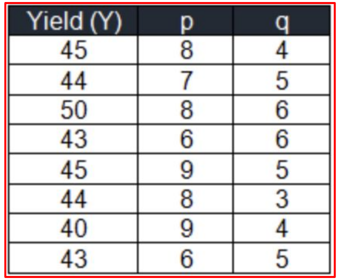
- 1. The relationship is linear
- 2. The errors have the same variance
- 3. The errors are independent of each other
- 4. The errors are normally distributed

Appendix

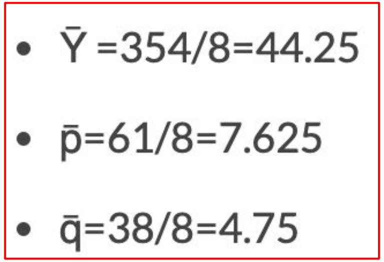
Feature Scaling



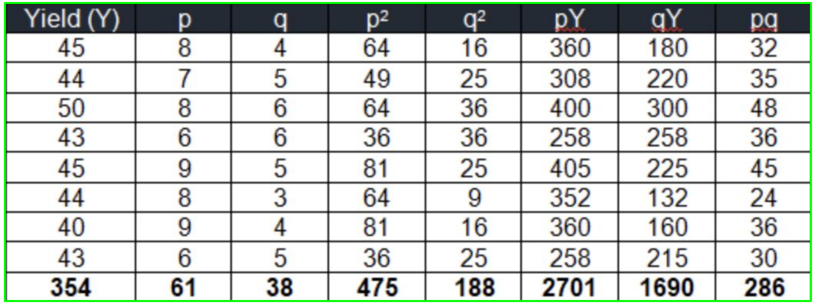
Multiple Linear Regression



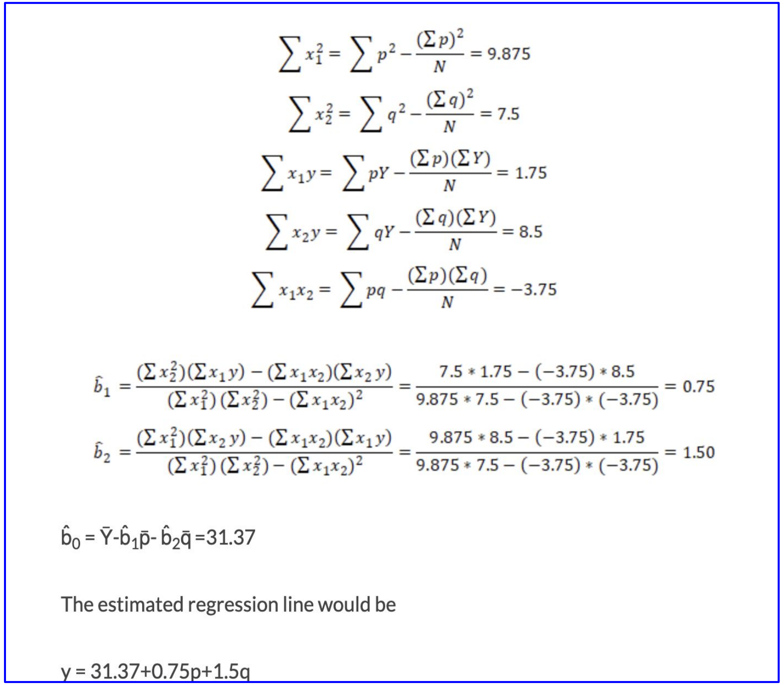
Yield (Y)	p	q
45	8	4
44	7	5
50	8	6
43	6	6
45	9	5
44	8	3
40	9	4
43	6	5



- $\bar{Y}$ =354/8=44.25
- $\bar{p}$=61/8=7.625
- $\bar{q}$=38/8=4.75



Yield (Y)	p	q	p²	q²	pY	qY	pq
45	8	4	64	16	360	180	32
44	7	5	49	25	308	220	35
50	8	6	64	36	400	300	48
43	6	6	36	36	258	258	36
45	9	5	81	25	405	225	45
44	8	3	64	9	352	132	24
40	9	4	81	16	360	160	36
43	6	5	36	25	258	215	30
354	**61**	**38**	**475**	**188**	**2701**	**1690**	**286**



$$\sum x_1^2 = \sum p^2 - \frac{(\sum p)^2}{N} = 9.875$$

$$\sum x_2^2 = \sum q^2 - \frac{(\sum q)^2}{N} = 7.5$$

$$\sum x_1 y = \sum pY - \frac{(\sum p)(\sum Y)}{N} = 1.75$$

$$\sum x_2 y = \sum qY - \frac{(\sum q)(\sum Y)}{N} = 8.5$$

$$\sum x_1 x_2 = \sum pq - \frac{(\sum p)(\sum q)}{N} = -3.75$$

$$\hat{b}_1 = \frac{(\sum x_2^2)(\sum x_1 y) - (\sum x_1 x_2)(\sum x_2 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)^2} = \frac{7.5 * 1.75 - (-3.75) * 8.5}{9.875 * 7.5 - (-3.75) * (-3.75)} = 0.75$$

$$\hat{b}_2 = \frac{(\sum x_1^2)(\sum x_2 y) - (\sum x_1 x_2)(\sum x_1 y)}{(\sum x_1^2)(\sum x_2^2) - (\sum x_1 x_2)^2} = \frac{9.875 * 8.5 - (-3.75) * 1.75}{9.875 * 7.5 - (-3.75) * (-3.75)} = 1.50$$

$\hat{b}_0 = \bar{Y} - \hat{b}_1\bar{p} - \hat{b}_2\bar{q} = 31.37$

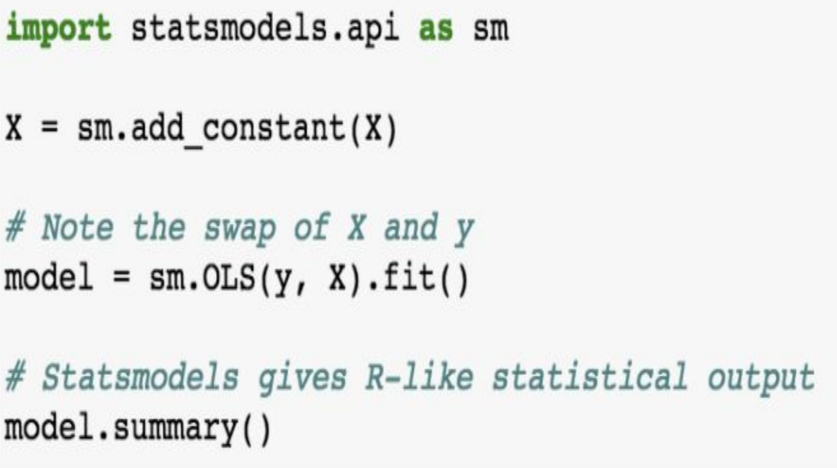
The estimated regression line would be

y = 31.37+0.75p+1.5q

Statsmodel vs. Sklearn

	Statsmodel	**Scikit-learn**
Model Evaluation	**ml.summary()** - p-values, CI; SE, beta ml.params()	**ml.score()** -prediction accuracy
Preferred Use	Statistical inference; hypothesis testing	Accurate prediction
Regularization	No regularization by default	L2 by default, optional (penalty = ‘none’)
Hyperparameter Tuning	User created	GridSearchCV
Intercept	includes by default	

Statsmodel

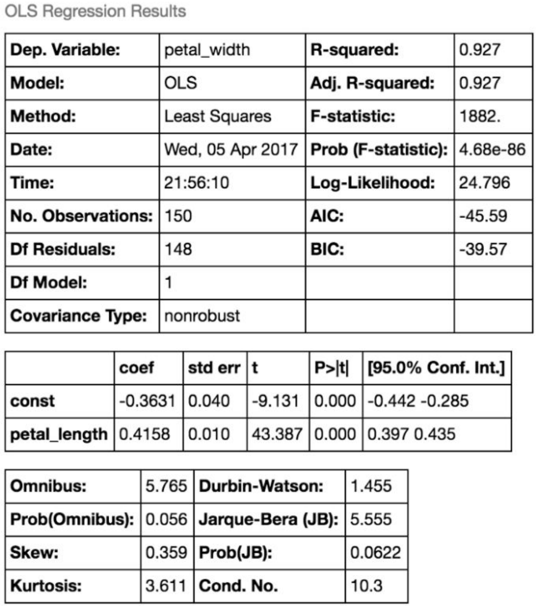


```
import statsmodels.api as sm

X = sm.add_constant(X)

# Note the swap of X and y
model = sm.OLS(y, X).fit()

# Statsmodels gives R-like statistical output
model.summary()
```



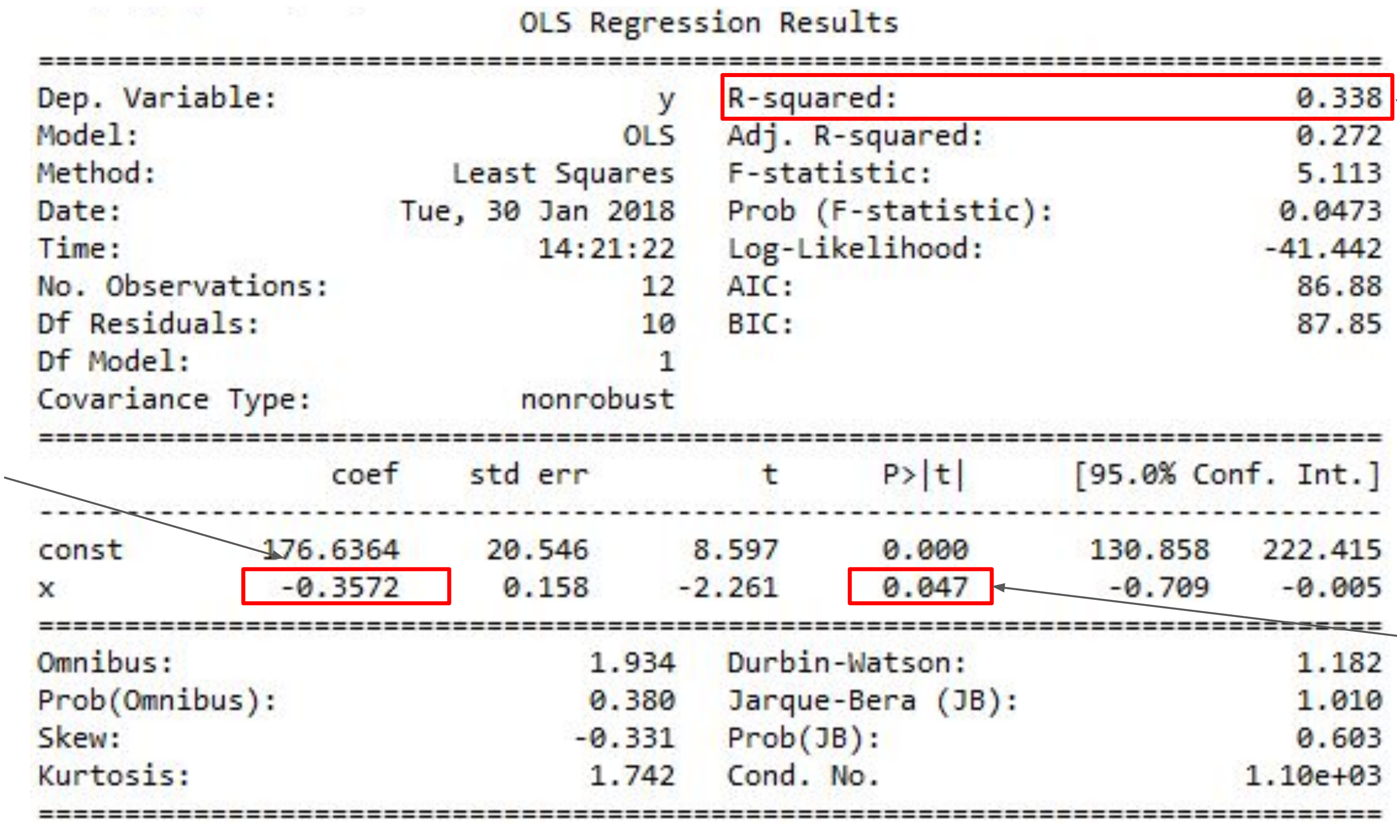
OLS Regression Results

Dep. Variable:	petal_width	**R-squared:**	0.927
Model:	OLS	**Adj. R-squared:**	0.927
Method:	Least Squares	**F-statistic:**	1882.
Date:	Wed, 05 Apr 2017	**Prob (F-statistic):**	4.68e-86
Time:	21:56:10	**Log-Likelihood:**	24.796
No. Observations:	150	**AIC:**	-45.59
Df Residuals:	148	**BIC:**	-39.57
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[95.0% Conf. Int.]**
const	-0.3631	0.040	-9.131	0.000	-0.442 -0.285
petal_length	0.4158	0.010	43.387	0.000	0.397 0.435

Omnibus:	5.765	**Durbin-Watson:**	1.455
Prob(Omnibus):	0.056	**Jarque-Bera (JB):**	5.555
Skew:	0.359	**Prob(JB):**	0.0622
Kurtosis:	3.611	**Cond. No.**	10.3

Interpreting Model Summary



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.338
Model:                            OLS   Adj. R-squared:                  0.272
Method:                 Least Squares   F-statistic:                     5.113
Date:                Tue, 30 Jan 2018   Prob (F-statistic):             0.0473
Time:                        14:21:22   Log-Likelihood:                -41.442
No. Observations:                  12   AIC:                             86.88
Df Residuals:                      10   BIC:                             87.85
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const        176.6364     20.546      8.597      0.000       130.858   222.415
x             -0.3572      0.158     -2.261      0.047        -0.709    -0.005
==============================================================================
Omnibus:                        1.934   Durbin-Watson:                   1.182
Prob(Omnibus):                  0.380   Jarque-Bera (JB):                1.010
Skew:                          -0.331   Prob(JB):                        0.603
Kurtosis:                       1.742   Cond. No.                     1.10e+03
==============================================================================
```

R^2: how well the model fits the data

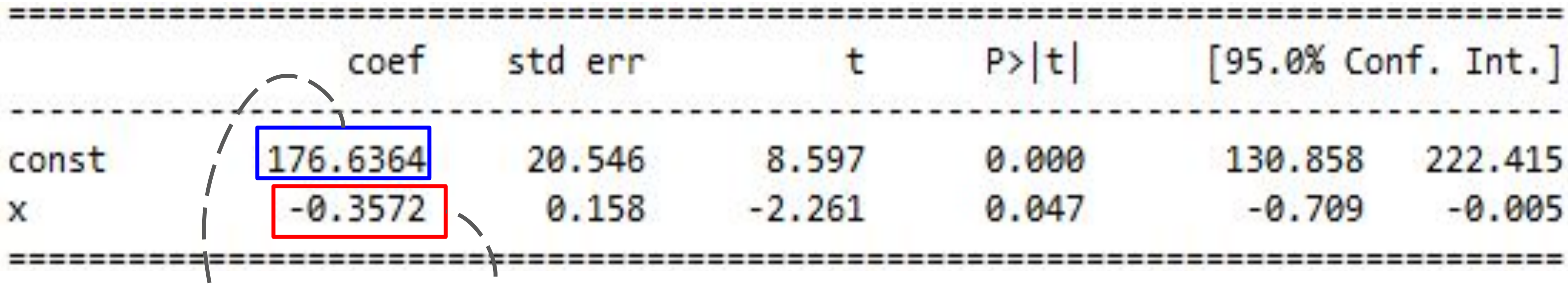
coefficient of the x (IV)

***p*-value: signif. of x (IV) for Y (DV)**

DV:dependent variable/outcome/target; **IV**: Independent variable, factor, predictor, feature

Linear Regression Formula

For every 1 unit change in x, the average change in y is b_1



```
==============================================================================
                 coef    std err          t      P>|t|    [95.0% Conf. Int.]
------------------------------------------------------------------------------
const        176.6364     20.546      8.597      0.000     130.858   222.415
x             -0.3572      0.158     -2.261      0.047      -0.709    -0.005
==============================================================================
```

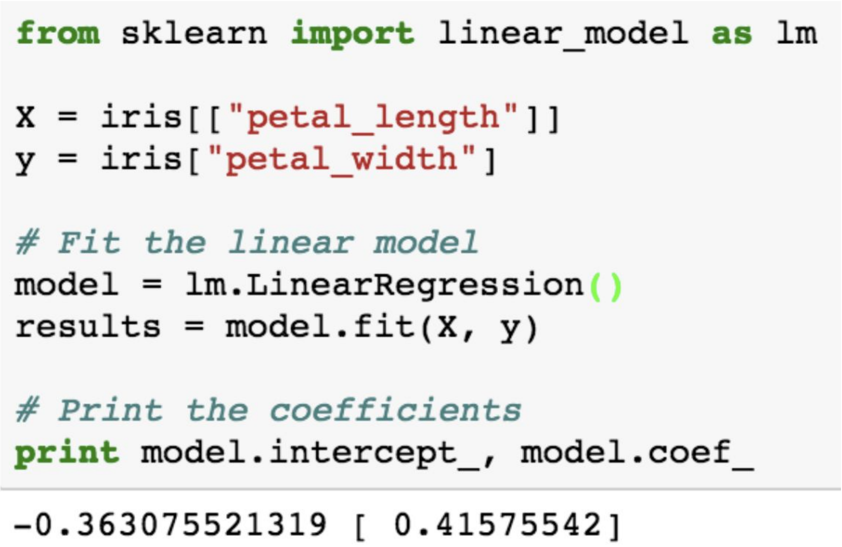
$$y = b_0 + b_1 x + e$$

y-intercept
(x=0)

slope of
regression

$$\boxed{y = 176.63 + -0.357x + e}$$

Sklearn



```
from sklearn import linear_model as lm

X = iris[["petal_length"]]
y = iris["petal_width"]

# Fit the linear model
model = lm.LinearRegression()
results = model.fit(X, y)

# Print the coefficients
print model.intercept_, model.coef_
```

```
-0.363075521319 [ 0.41575542]
```